REMARKS

The original application, filed October 31, 2003, included claims 1-35. An Office action of June 14, 2007, presented a restriction requirement with claims grouped as claims 1-11, 13-23, and 25-34 in Group I, and claims 12, 24, and 35 in Group II. In a Response to Restriction Requirement of July 6, 2007, Applicant elected to prosecute the claims in Group I, claims 1-11, 13-23, and 25-34, without traversal, wherein the claims of Group I were to be prosecuted alone (and the claims of Group II were to be correspondingly withdrawn).

In a nonfinal Office action of September 27, 2007, Examiner objected to claims 2, 14, 26 for minor informalities. Examiner also objected to claims 6-8 on grounds that they had improper dependency numbering.

In Reply A, filed December 27, 2007, Applicant responsively amended claims 2, 14 and 26 in accordance with Examiner's request. Applicant also responsively amended claim 1 to incorporate claims 6 and 8, canceled claims 6 and 8, and amended claim 7, thereby overcoming the objection.

A final Office of March 3, 2008, presented new grounds of rejection. Specifically, claims 1, 3-11,13,15-23, 25 and 27-34 were rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent Publication No. 200310101353 ("Tarquini") in view of US Patent No. 7,185,368 ("Copeland"), or possibly US Patent No. 6,851,061 ("Holland "). (Copeland was recited in the broad statement of the rejections, but Holland is recited in the detailed remarks.) Claims 2, 14, and 26 were rejected under 35 U.S.C. 103(a) as being unpatentable over Tarquini in view Copeland, or possibly Holland. (Again, Copeland was recited in the broad statement of the rejections, but Holland was recited in the detailed remarks.) Claims 11, 23, and 34 were rejected under 35 U.S.C. 103(a) as being unpatentable over Tarquini in view of US Patent Publication No 2004101 17478 ("Triulzi").

In a Reply accompanying a Request for Continued Examination, Applicant traversed the rejection and also submitted amendments to claims 1, 2, 7, 13, 14, 19, 20, 25, 26, and 31 and submitted new claims 36-44, merely to expedite allowance.

The present, non-final Office action of October 17, 2008 (the "present Office action"), withdrew the rejection of claims 1-4,7,9- 11,13-16, 19-23, 25-28, 31 and 33-34.

However, the present Office action introduced new grounds of rejection. Claims 1-4, 7, 9-11, 13-16, 19-23, 25-28, 31, 33, 34 and 36-44 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Yadav (US Patent No 7174566) in view of Holland (US Patent No 6,851,061).

Intrusion signatures are often piecemeal. That is, a network intrusion can be camouflaged in different network packets that can cause a problem when coalesced. Intrusion detection between the transport and network layers does not detect signatures spread across packets, since the network layer doesn't have the ability or the knowledge to coalesce fragmented packets. Coalescing such fragmented packets is the job of the transport layer. The application layer needs to be presented information as a whole and not in the fragmented form in which the packets arrived. It is, therefore, evident that scanning is more effective according to the arrangement recited in the claims of the present application.

Regarding claim 1, the Office action asserts that Yadav, column 5, lines 14-32, teaches "a method of detecting an intrusion in a communications network, the method comprising the steps of: a) accessing, by a network intrusion detection process of a target computer system, communication to an application receive queue (ARQ) for an application running in an application layer of the target computer system, wherein the ARQ functions intermediate the application layer and a transport layer of a network protocol associated with said communications network to receive data packets for the application from the transport layer;" and asserts that Yadav, column 6, lines 17-37, teaches "b) scanning for the application by the network intrusion detection process only the data packets accessed by the network intrusion detection process in a), wherein the data packets are directed to the application from a remote host via the communications network, and wherein the scanning is after the data .packets have been processed by the e-transport layer and after the transport layer has passed the processed data packets for receipt by the application's ARQ," as recited by claim 1 in the present application. Applicant respectfully disagrees for at least the following reasons.

Yadav teaches the use of an intrusion detection system 230 between a network driver and a transport layer. Yadav, col. 5, lines 1-4. Yadav states that IDS 230 may also have "additional components 232 placed elsewhere in the network stack." Yadav, col. 5, lines 7-8. Yadav goes on to state that "application-level detection may be implemented in one or more components placed just below and/or just inside the application layer 220." Yadav, col. 5, lines 10-13. (Yadav explains that this can be done by an application-level component 234 and/or individual application-level components 236 for respective applications. Yadav, col. 5, lines 14-15 and 25-26.)

However, Yadav makes it clear that these "one or more components placed just below and/or just inside the application layer 220" are in addition to IDS 230. See 5, lines 14-15 (stating that component 234 is "part of" IDS 230); see also Yadav, col. 5, lines 25-26 (stating that components 236 may be in addition to component 234); see also Yadav, col. 5, lines 25-26 (stating that components 236 may be an alternative to component 234, and not stating that component 234 is an alternative to 230). That is, Yadav teaches that IDS 230 performs a firewall function that includes monitoring network traffic to block traffic that is a prelude to intrusion. Yadav, col. 5, lines 42-53. Nowhere does Yadav teach or suggest that application-level components 234 or 236 perform the monitoring and blocking. And nowhere does Yadav teach or suggest that the basic IDS 230 is located somewhere other than between the network driver and the transport layer.

Thus, it should be appreciated from the above that in connection with intrusion detection system 230 and application-level components 234 and 236, Yadav does not teach or suggest scanning of packets that have been processed by the transport layer and are on their way to a particular application receive queue. See claim 1 of the present case ("a) accessing . . . communication to an application receive queue (ARQ) . . . , wherein the ARQ functions intermediate the application layer and a transport layer . . . to receive data packets for the application from the transport layer" and "scanning for the application . . . only the data packets accessed by the network intrusion detection process in a), . . . wherein the scanning is after the data packets have been processed by the transport layer and after the transport layer has passed the

processed data packets for receipt by the application's ARQ"). Claims 13 and 25 have similar language, according to the forms of the invention they claim.

That Yadav does not teach or suggest what is claimed in the present case is made all the more clear by analysis of teaching by Yadav in connection with FIG's 2B, 3 and 4 at col. 5, line 35 – col. 8, line 33. That is, Yadav teaches that a structure illustrated in FIG. 2B includes a network traffic enforcer 282, which, like IDS 230 of FIG. 2A, is shown below a transport layer. Further, Yadav's FIG. 2B illustrates an application rule enforcer 284 below an application layer, like application level component 234 of FIG. 2A. In the description of the processes illustrated in FIG's 3 and 4, Yadav makes more clear how the component below the application layer communicates with the component below the transport layer, and explains how the lower component monitors and blocks. From this explanation, it is clear that what Yadav teaches is very different than the scanning claimed in the present case.

In particular, Yadav teaches that application rule enforcer 284 handles an outgoing network service request from an application. Yadav, col. 7, lines 53-60. If the request is within the permitted policy, application rule enforcer 284 signals network traffic enforcer 282 to open a "channel" for the application, for which Yadav describes an example. Yadav, col. 7, lines 53-60 (describing how channel is defined by application rule enforcer specifying a channel protocol, source and destination IP addresses and source and destination ports). Network traffic enforcer 282 responsively opens the channel and adds it to an authorization list 405. Yadav, col. 8, lines 21-24. Further, "The network traffic enforcer 282 monitors incoming network traffic. If an inbound communication 262 fails to correspond to an authorized request (i.e., the inbound communication was not effectively pre-approved by the application rule enforcer), the communication is dropped (i.e., blocked from passage to another layer in the network stack.)." Yadav, col. 6, lines 26-31. Yadav does not explicitly state how network traffic enforcer 282 determines that an inbound communication 262 fails to correspond to an authorized request. But it would be logical, in view of what Yadav does explicitly teach, for this to be done by matching an authorized channel on the authorization list 405 with a "channel" indicated by inbound communication 262. This would, of course, be done below the transport layer, since it would be done by the

network traffic enforcer 282. See Yadav, col. 6, lines 26-31 ("The network traffic enforcer 282 monitors incoming network traffic. If an inbound communication 262 fails to correspond to an authorized request . . . the communication is . . . blocked from passage to another layer . . .").

Thus, it should be appreciated even more particularly from the above that in connection with network traffic enforcer 282 and application rule enforcer 284, Yadav does not teach or suggest scanning of packets that have been processed by the transport layer and that are on their way to a particular application receive queue. Again, see claim 1 of the present case ("a) accessing . . . communication to an application receive queue (ARQ) . . . , wherein the ARQ functions intermediate the application layer and a transport layer . . . to receive data packets for the application from the transport layer" and "scanning for the application . . . only the data packets accessed by the network intrusion detection process in a), . . . wherein the scanning is after the data packets have been processed by the transport layer and after the transport layer has passed the processed data packets for receipt by the application's ARQ"). Claims 13 and 25 have similar language, according to the forms of the invention they claim.

For all the above reasons, Applicant respectfully submits that claims 1, 13 and 25 are patentably distinct.

Regarding claims 2, 14 and 26, the Office action points out that Yadav column 5, lines 47-52, teaches tracking abnormally behaving applications. The Office action states that "it is common knowledge that if an application is behaving abnormally, and that it is possible due to an intrusion, then that application should be terminated in order to prevent any harmful effects that may result from the intrusion" and that it would have been obvious for one of ordinary skill in the art to modify Yadav to include terminating the application responsive to determining a scanned data packet is malicious, in order to prevent any harmful effects that may result from the intrusion. Applicant respectfully submits that this analysis is not relevant to the present invention, as claimed.

In the present case, intrusions are detected before data received from a remote host for an application has been passed to the application, and thus before the data can cause the application to behave abnormally by scanning data for an application in the manner recited in claim 1. See previous explanation on page 13 of Applicant's Response Accompanying Request for Continuing Examination, submitted August 4, 2008. Nevertheless, in one aspect of the present invention, as recited in claim 2 when read in connection with claim 1, an application is terminated responsive to detecting the malicious data packet. It logically follows from claim 1 and 2 of the present case that an application is terminated due to detecting an intrusion, but before the application can be subjected to any harmful effects of the intrusion and, therefore, before the application has an opportunity to behave abnormally. This does not follow from the logic presented in the Office action that posits reasons for terminating an application if the application is behaving abnormally due to an intrusion. Therefore, it would not be obvious to modify Yadav according to the logic put forward by the Office action. Consequently, Applicant submits that claims 2, 14 and 26 are patentably distinct.

Further, an Examiner may rely on facts within his or her own knowledge to support a rejection. On the other hand, however, the Applicant is entitled to an affidavit or declaration from the Examiner setting forth specific factual statements and an explanation to support the finding, provided that the Applicant requests the affidavit or declaration. MPEP 2144.03(C). Accordingly, Applicant respectfully requests that if the present rejection of claims 2, 14 and 26 is maintained Examiner submit an examiner's affidavit or declaration providing a basis for alleged common knowledge presented in the Office action.

Regarding claims 37, 40 and 43, the Office action asserts that Yadav, column 6, lines 25-31, teaches intimating a transport layer to tear down a remote host connection responsive to determining a scanned data packet is malicious. Applicant respectfully disagrees. In this passage, Yadav teaches that "If an inbound communication 262 fails to correspond to an authorized request . . . the communication is dropped (i.e., blocked from passage to another layer . . ." Blocking communication from passing to another layer is not the same as, nor does it suggest, intimating the transport layer to tear down

a remote host connection responsive to determining a scanned data packet is malicious. Tearing down a connection typically includes invoking a "send disconnect" type function and also invoking a "close socket" type function, for example. See, for example, http://msdn.microsoft.com/en-us/library/ms737616(VS.85).aspx. This closes a channel of communication that existed between the application and the remote host via the network and does not merely block communication from passing to another layer in the target computer system. Applicant submits, therefore, that claims 37, 40 and 43 are patentably distinct.

Regarding method claims 3, 4, 7, 9-11, 36 and 38, system claims 15, 16, 19-23, and 39 and 41, and computer program product claims 27, 28, 31, 33, 34, 42 and 44, Applicant submits that these claims are allowable at least because they depend upon allowable base claims.

REQUESTED ACTION

Applicant submits that the claims as submitted herein are patentably distinct, and hereby requests that Examiner grant allowance and prompt passage of the application to issuance.

Respectfully submitted,

Anthony V. S. England
Attorney for Applicant
Registration No. 35,129
512-477-7165
a@aengland.com